



UNIVERSIDADE
DE VIGO

ESCOLA SUPERIOR DE ENXEÑARÍA

Memoria do Traballo de Fin de Grao que presenta

D. Julián Jiménez González

para a obtención do Título de Graduado en Enxeñaría Informática

SuperCars: Desarrollo de videojuego para dispositivos móviles



Xaneiro, 2015

Traballo de Fin de Grao N°: EI 14/15 - 010

Titor/a: Alma María Gómez Rodríguez

Área de coñecemento: Linguaxes e Sistemas Informáticos

Departamento: Informática

Índice

1.	DESCRIPCIÓN DEL PROYECTO	3
1.1.	INTRODUCCIÓN	3
1.2.	OBJETIVOS	4
1.3.	DOCUMENTACIÓN ENTREGADA	4
1.3.1.	<i>Memoria de Trabajo</i>	4
1.3.2.	<i>Manual de Usuario</i>	4
1.3.3.	<i>Manual Técnico</i>	4
1.3.4.	<i>Código fuente</i>	4
2.	TECNOLOGÍAS, METODOLOGÍAS Y HERRAMIENTAS	5
2.1.	COCOS2D-X V3.2 (C++)	5
2.2.	ANDROID NDK V.R9D	5
2.3.	PYTHON V2.7.8	6
2.4.	APACHE ANT V1.9.4	6
2.5.	ECLIPSE ADT BUNDLE	6
3.	PLANIFICACIÓN TEMPORAL	6
4.	SEGUIMIENTO	7
4.1.	REPARTO TEMPORAL REAL	7
4.2.	FUNCIONALIDADES	7
4.2.1.	<i>Sprint 1. Implementación de los controles del vehículo</i>	8
4.2.2.	<i>Sprint 2. Adición de circuito y colisiones</i>	8
4.2.3.	<i>Sprint 3. Creación de menús</i>	8
4.2.4.	<i>Sprint 4. Incorporación de contrincantes</i>	8
4.2.5.	<i>Sprint 5. Modificaciones de vehículo</i>	8
4.2.6.	<i>Sprint 6. Mejoras y pruebas globales</i>	8
4.3.	DESVÍOS	8
4.3.1.	<i>Implementación de Box2D</i>	8
5.	PROBLEMAS ENCONTRADOS Y SOLUCIONES APORTADAS	9
5.1.	IMPLEMENTACIÓN DE BOX2D	9
6.	CONCLUSIONES	9
7.	MEJORAS Y POSIBLES AMPLIACIONES	9
8.	REFERENCIAS	9

1. Descripción del Proyecto

1.1.Introducción

Desde que se comenzaron a comercializar en la década de los 90, el concepto de "videojuego" no ha hecho más que evolucionar hasta lo que conocemos hoy en día, llegando incluso en algunos países al estatus de medio artístico.

Así ha sido en EEUU, donde el Museum of Modern Art (MoMA) ya ha introducido un buen número de videojuegos en sus colecciones.

Tanto es así que, según el informe anual de 2014 de NewZoo (*Ilustración 1*), la industria del videojuego generó más de \$80 billones y cuenta con más de 1700 millones de jugadores en todo el Mundo.

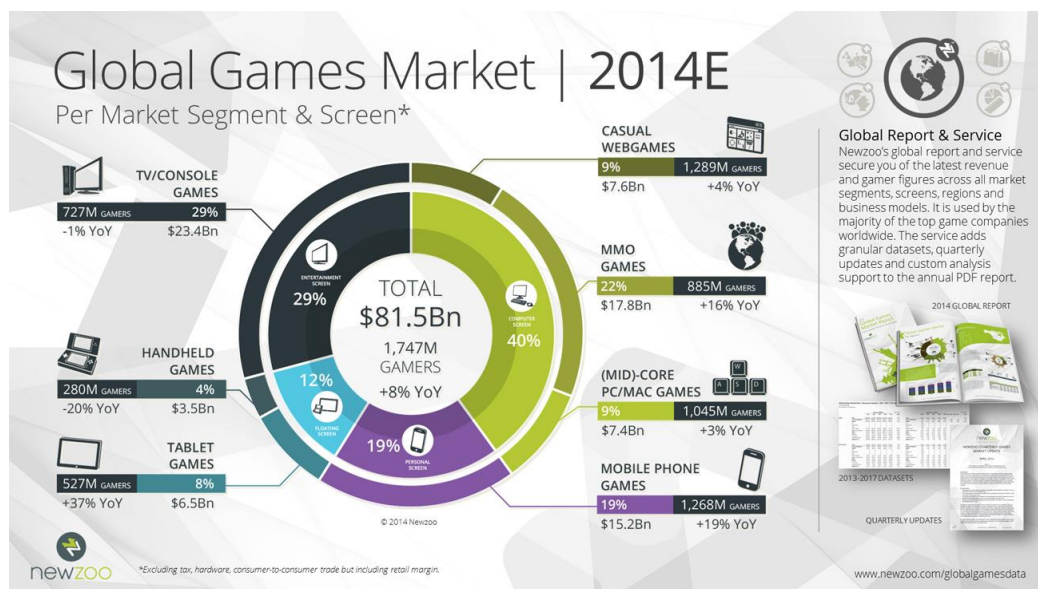


Ilustración 1

Se considera que en 2015 se podrían alcanzar los \$100 billones, poniéndose a la altura de la música y el cine.

Esta situación favorece la aparición de cada vez más salidas profesionales en el sector, además de mucha documentación y herramientas que permiten a cualquier persona interesada desarrollar su propio videojuego.

Entre las distintas plataformas de videojuegos, la de dispositivos móviles (Smartphones y Tablets) destaca como la de mayor auge en la actualidad, y (como refleja la *Ilustración 1*) la que mayor crecimiento anual prevé.

Por su parte, el SO Android cuenta con una cuota del mercado global de smartphones y tablets de alrededor del 85%, siendo videojuegos más del 40% de las aplicaciones descargadas para estos dispositivos. Esto puede relacionarse directamente con el creciente interés de las personas por los juegos "casual", a

los que no necesitan dedicar mucho tiempo y que les permiten distraerse durante unos minutos de sus actividades cotidianas. Además, el reducido coste de estos videojuegos para plataformas móviles los hace mucho más accesibles para todo el mundo.

Es por esto que en este proyecto se ha decidido realizar un videojuego para Android.

1.2. Objetivos

El objetivo de este proyecto es desarrollar un videojuego de carreras en 2D de un jugador para Android.

Desde los menús se podrán realizar las configuraciones mínimas de la carrera y competidores (número de vueltas, dificultad de los contrincantes, etc...).

Durante la carrera se tendrá una vista aérea y se podrán ver los contadores de tiempo, vueltas y posición en carrera. Los rivales tendrán distintos niveles de dificultad. Esto influirá en los parámetros de su vehículo o su comportamiento durante la carrera.

Además de la línea del circuito, habrá zonas que modifiquen el comportamiento del vehículo (como arena o hierba) y muros que lo limiten.

También se guardará un ranking de vueltas rápidas.

1.3. Documentación entregada

La documentación correspondiente a este TFG es la siguiente:

1.3.1. Memoria de Trabajo

Este mismo documento, en el cual se recoge toda la información referente a la planificación y gestión del proyecto, así como los desvíos producidos en tiempo y forma con respecto a dicha planificación.

1.3.2. Manual de Usuario

Documento en el que se describe de forma visual cómo funciona la aplicación de cara a los usuarios finales de la misma.

1.3.3. Manual Técnico

En este documento se detalla la aplicación desde el punto de vista de un desarrollador, facilitando las tareas de mantenimiento y ampliación de la misma.

1.3.4. Código fuente

En un CD se adjuntará el código fuente de la aplicación con la estructura siguiente:

- **/codigo**. Directorio que contendrá los archivos del código fuente.
- **SuperCars.apk**. Archivo de instalación de la aplicación para dispositivos Android.

2. Tecnologías, metodologías y herramientas

Teniendo en cuenta la inexperiencia del desarrollador con el entorno de desarrollo y las tecnologías utilizadas, para este proyecto se ha considerado que un desarrollo iterativo e incremental (inspirado en SCRUM) es lo más adecuado, permitiendo adaptarse a posibles cambios o problemas que surjan durante el proyecto.

Se ha realizado, por tanto, un desarrollo basado en sprints que añaden funcionalidad de forma iterativa e incremental a la aplicación.

Todos estos sprints contienen distintas tareas, citadas a continuación:

- Análisis del alcance y diseño de la funcionalidad que se va a desarrollar.
- Implementación.
- Pruebas de unidad.
- Documentación.

La división prevista en sprints es la siguiente:

- Sprint 1. Implementación de los controles del vehículo.
- Sprint 2. Adición de circuito y colisiones.
- Sprint 3. Incorporación de contrincantes.
- Sprint 4. Creación de menús.
- Sprint 5. Modificaciones de vehículo.
- Sprint 6. Mejoras y pruebas globales.

A continuación se listan las tecnologías y recursos empleados, así como su utilidad para el desarrollo del proyecto:

2.1.Cocos2d-x v3.2 (C++)

Tras estudiar otras posibilidades como Unity3D, AndEngine o LibGDX, este ha sido el framework seleccionado para desarrollar el proyecto.

Las razones principales son dos:

Es una herramienta open-source y gratuita, tanto para proyectos comerciales como no comerciales.

Posee una documentación bastante completa para iniciar el desarrollo, así como una comunidad muy activa y amplia.

Además, Cocos2d-x emplea C++ como lenguaje de desarrollo, siendo el más extendido en el desarrollo profesional de videojuegos y que ofrece un gran rendimiento en aplicaciones que precisen cargas de trabajo intensas de la CPU.

2.2.Android NDK v.r9d

Es una herramienta que permite implementar partes de una aplicación Android en código nativo (como C++), a cambio de un aumento en la complejidad de dicha aplicación. Lo que hace es que la ejecución de la aplicación sea directamente sobre el procesador y no interpretada por una máquina virtual.

Si bien no es recomendable para cualquier aplicación, sí tiene cabida en aplicaciones que, como se ha comentado anteriormente, requieren una carga de

trabajo importante de la CPU. Además de ser necesaria alguna herramienta que permita compilar código C++ en Android.

NOTA: En el momento en que se inicia el desarrollo, la versión r9d es la óptima para trabajar con Cocos2d-x, no siendo válidas versiones superiores.

2.3. Python v2.7.8

A partir de Cocos2d-x 3.0, éste emplea Python para configurar y compilar los archivos de la aplicación.

NOTA: debido a ciertos cambios en la sintaxis de Python, Cocos2d-x no es compatible con Python v3.0+.

2.4. Apache Ant v1.9.4

Esta herramienta se utiliza para la automatización de procesos de compilación y construcción.

Su función es similar a la de Make, pero no depende de las órdenes del Shell de cada SO (gracias a la Java Virtual Machine) y es más eficaz a la hora de determinar qué archivos están actualizados.

2.5. Eclipse ADT Bundle

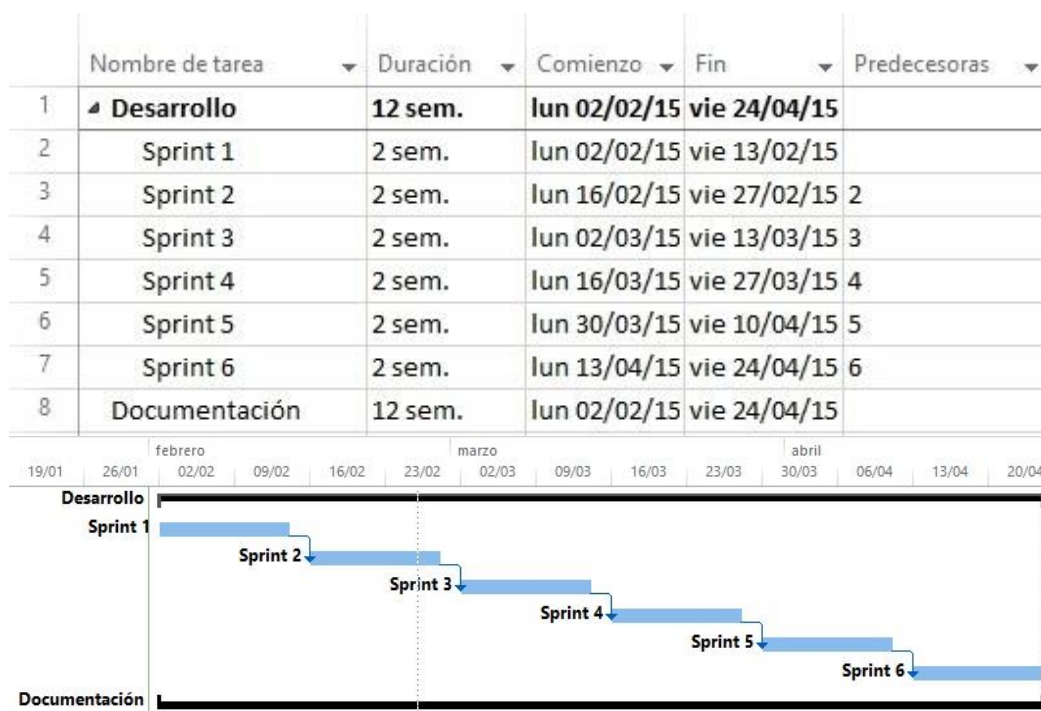
Éste paquete contiene el IDE Eclipse junto a Android SDK (ya configurado), por lo que es una opción muy recomendable para desarrollar aunque no imprescindible.

3. Planificación temporal

La planificación inicial para la distribución de horas es la siguiente:

Dedicación semanal prevista (en horas/semana): 25	
Fase	Estimación temporal (en semanas)
Sprint 1	2
Sprint 2	2
Sprint 3	2
Sprint 4	2
Sprint 5	2
Sprint 6	2
TOTAL PROXECTO	12

A continuación se muestra el Diagrama de Gantt que representa la disposición de horas a lo largo del tiempo de desarrollo estimado.



Nótese que este método de desarrollo es susceptible a cambios durante el mismo.

4. Seguimiento

A pesar de la planificación, no ha sido posible mantener un ritmo constante de trabajo durante todo el periodo de desarrollo, por lo que se han producido desvíos en los plazos de finalización de cada sprint.

Éste ha sido el reparto real de horas del proyecto. Posteriormente se detallarán ciertos desvíos (4.3) causados por los problemas encontrados durante el desarrollo.

4.1. Reparto temporal real

FASE	TIEMPO (HORAS)
SPRINT 1	87,27
SPRINT 2	42,65
SPRINT 3	12,71
SPRINT 4	
SPRINT 5	
SPRINT 6	
TOTAL	142,63

4.2. Funcionalidades

A continuación se detallan las funcionalidades desarrolladas en cada Sprint, éstas han sido decididas al inicio de los mismos, en función de:

- Funcionalidades ya implementadas.

- Funcionalidades restantes.
- Tiempo de desarrollo real con respecto al estimado.

4.2.1. Sprint 1. Implementación de los controles del vehículo

- Escena principal de carrera.
- Objeto jugador.
- Botones para controlar al jugador y eventos asociados a éstos.
- Mapa simple (estático).

4.2.2. Sprint 2. Adición de circuito y colisiones

- Mapa TMXTiled, dinámico e infinito.
- Creación aleatoria de obstáculos y movimiento de los mismos.
- Control de colisiones entre jugador y obstáculos.
- Música de fondo y efectos sonoros básicos.
- Creación de líneas de vuelta y conteo de las mismas.

4.2.3. Sprint 3. Creación de menús

- Creación de todos los menús del juego (y botones correspondientes):
 - Menú principal.
 - Menú de configuración de carrera.
 - Menú de pausa durante la carrera.
 - Menú de fin de carrera.
- Adición de GUI (Graphic User Interface) durante la carrera.

4.2.4. Sprint 4. Incorporación de contrincantes

4.2.5. Sprint 5. Modificaciones de vehículo

4.2.6. Sprint 6. Mejoras y pruebas globales

4.3. Desvíos

En este apartado se detallan y justifican los desvíos producidos con respecto a la planificación inicial.

4.3.1. Implementación de Box2D

El problema de implementación del motor de físicas Box2D provocó un desvío aproximado de:

TAREA	DESVÍO (HORAS)
INVESTIGACIÓN	20
DESARROLLO	10
BÚSQUEDA DE RAZONES Y SOLUCIONES CORRESPONDIENTES	20
IMPLEMENTACIÓN DE SOLUCIÓN FINAL	10
TOTAL	60,00

5. Problemas encontrados y soluciones aportadas

A continuación se detallan los problemas y dificultades surgidos durante el desarrollo del proyecto, así como las soluciones adoptadas.

5.1. Implementación de Box2D

Inicialmente se decidió tratar de implementar el vehículo utilizando el motor de físicas Box2D, ya que parecía ser a priori la solución idónea.

No se consiguió que dicho motor de físicas funcionase en el proyecto para obtener el resultado esperado. A pesar de buscar una razón que lo justificase, no se ha podido confirmar ninguna de las posibles razones como causante del problema. Las posibilidades que se barajaron tras muchas horas de investigación son:

- **Tamaño de las imágenes mayor del soportado por el dispositivo.** Se emplearon las mismas imágenes en las versiones con y sin Box2D, por lo tanto:
 - **Descartado**
- **Errores en el código fuente.** Se encontró una versión de ejemplo similar a lo esperado en un repositorio público de la plataforma GitHub. Se probó el código fuente público y posteriormente, ya que no funcionaba, se contactó con el propietario del repositorio. Éste envió por email otra versión que sí debería funcionar, pero el resultado fue el mismo. Por lo tanto:
 - **Descartado**
- **Incompatibilidad del Hardware disponible.** Es posible que el recurso Hardware empleado para desarrollar la aplicación no soporte la inclusión de Box2D. Por ello se probó en otro modelo, en el que tampoco se podía arrancar la aplicación. Por lo tanto:
 - **No demostrable**

Finalmente, se decidió que el desvío comenzaba a ser elevado y habría que buscar una solución alternativa.

Solución: Simplificar las mecánicas del juego y eliminar Box2D del desarrollo.

6. Conclusiones

7. Mejoras y posibles ampliaciones

8. Referencias
